



FStream: a decentralized and social music streamer

Antoine Boutet, Konstantinos Kloudas, Anne-Marie Kermarrec

► To cite this version:

Antoine Boutet, Konstantinos Kloudas, Anne-Marie Kermarrec. FStream: a decentralized and social music streamer. NETYS, May 2013, Marrakech, Morocco. hal-00828542

HAL Id: hal-00828542

<https://inria.hal.science/hal-00828542>

Submitted on 31 May 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

FStream: a decentralized and social music streamer

Antoine Boutet¹, Konstantinos Kloudas¹, and Anne-Marie Kermarrec^{1,2}

INRIA Rennes, France

`antoine.boutet@inria.fr`, `konstantinos.kloudas@inria.fr`,

`anne-marie.kermarrec@inria.fr`

EPFL, Switzerland

`anne-marie.kermarrec@epfl.ch`

Abstract. Internet streaming services and social networks have drastically changed how people discover and consume music. Existing streaming services allow users to listen to music available on a centrally controlled web infrastructure. However, recent trends, *e.g.* large, inexpensive home storage devices and always on, high-speed broadband connectivity, provide the opportunity for users to collaboratively share their music collections directly, without the intervention of a service provider. In this paper we present FSTREAM, a distributed, social music streaming service. Users contribute to the system by hosting at home a FSTREAM-BOX which enables them to have access to their music collection everywhere and at any time. In addition, they can share their collection with friends, discover new tracks thanks to a recommendation system and search for specific tracks in the whole network without using a central entity. FSTREAM provides a fine grained sharing policy allowing the user to have full control over the way she shares her music and manage resource allocation to address traffic spikes and content availability.

Keywords: Social Networks, Recommendation System, Streaming

1 Introduction

In recent years the way we are consuming music has drastically changed. Aided by technological advances in network access and storage, personal music collections have grown considerably while, at the same time, fit in devices as small as a usb stick. The above, in combination with the integration of music players in mobile devices like smartphones that we carry everywhere, has led to music accompanying many of our everyday activities. The above phenomenon has led to the emergence of numerous Internet services like Spotify ¹ or Jango ² that provide large collections of tracks directly available for streaming (without downloading).

¹ <http://www.spotify.com>

² <http://www.jango.com>

An aspect of music that does not go unnoticed by these services is the social one. Music sharing sites, in their effort to improve their offered services and to engage users, add more and more functionalities based on user interactions. This includes music recommendation based on collaborative filtering, track rating and playlist sharing. On the other hand, and for the same reasons, social networking sites tend to integrate music sharing functionalities. The recent deal between Facebook and Spotify illustrates the above.

Existing services, allow users to listen to music available on a centrally controlled web infrastructure. Even services like Spotify that are peer-assisted [4], assume the existence of a central index, which serves as the gateway to locate music tracks. However, recent trends such as large, inexpensive home storage devices and always on, high-speed broadband connectivity at low (and decreasing) cost provide the opportunity for users to collaboratively share their music collections directly from home [6, 5], without the intervention of a service provider. Hosting a streaming server at home, removes the need for users to install specific software on each one of their devices or download each track before listening to it, as in existing filesharing applications. In addition, an alternative architecture that permits the direct interaction between users, avoids drawbacks of the centralized alternatives: (i) no site-specific constraints on the upload data (e.g. format, size, etc) and no need to upload content to remote data centers; (ii) avoid complex terms of ownership rights (e.g. facebook, instagram) and let users regain control over their data (i.e., what to share, whom to share with); (iii) no advertisement and service restriction (e.g. restricted free access) or restriction on the available music collection (e.g. only albums from one label); (iv) scalability issue (e.g. track unavailable due to social sprike).

In this paper we present FSTREAM, the first distributed, social music streaming network. FSTREAM enables users to have access to their own music collection everywhere and at any time. In addition, they can share their collection with friends, discover new tracks thanks to a recommendation system that brings closer people with similar music tastes, and search specific tracks in the whole network without relying on a central entity. FSTREAM provides a fine grained sharing policy that permits the user to have full control over the way she shares her music. Instead of installing a p2p client, users in FSTREAM use a browser to connect their box called FSTREAMBOX. FSTREAM is hosted at home and takes care of managing all operations for them. In particular, FSTREAM helps users to maintain their social network, compute recommendation based on their music tastes and manage resource allocation to address traffic spikes and content availability.

2 Overview of FStream

FSTREAM resides on user provided resources leveraging their capabilities in storage and bandwidth. In FSTREAM, each user contributes to the system by hosting at home a FSTREAMBOX, which grants access to not only its own music collection but the ones of other members of the service. The only thing required to

benefit from the streaming service is a web browser to connect to her machine. All the operations of referencing friends' collections, discovering new nodes and music collections, computing recommendations and content caching are managed by the FSTREAMBOX. We based our solution on Subsonic³, an open-source, web-based media streamer and added several components implementing the additional functionality provided by FSTREAM. This additional functionality includes: (i) social network construction and maintenance, (ii) ownership and sharing restrictions control, (iii) recommendation system, (iv) search mechanism, (v) storage space and social caching.

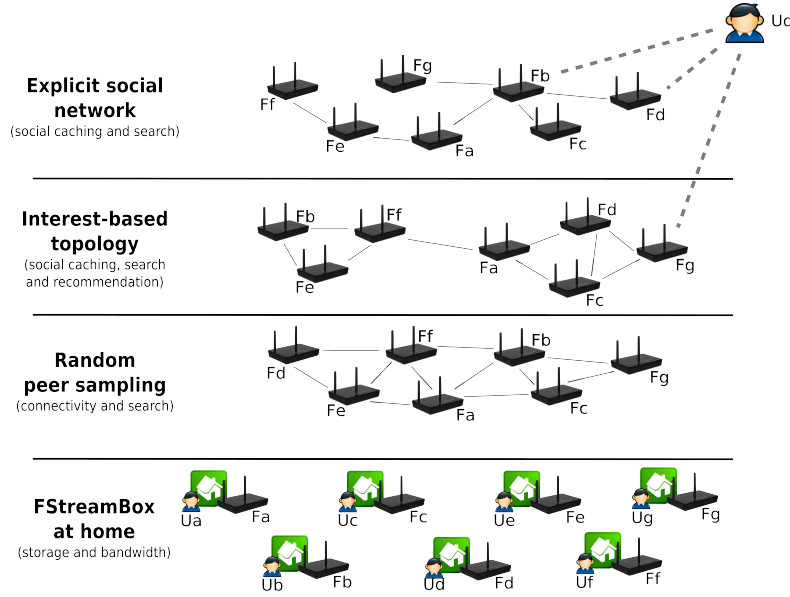


Fig. 1. FSTREAM architecture.

Figure 1 depicts the architecture of a FSTREAM system. Our system can be seen as composed of four different layers. At the bottom layer are the FSTREAMBOXES that provide the storage service and are the link between a user and the rest of the system. The second layer relies on a random-peer-sampling (rps) protocol [7] which ensures connectivity between FSTREAMBOXES by building and maintaining a continuously changing random topology. This random overlay is also leveraged to search items through the system. The upper-layer clustering protocol [8] uses this overlay to provide users with the most similar candidates based on their profiles (*i.e.* short descriptors of their tastes) to form an interest-based topology. This gossip-based clustering protocol, is based on a periodic exchanges of user profiles, on which it applies a similarity metric to cluster users

³ <http://www.subsonic.org>

with similar music tastes. To guarantee that this overlay represents the user's most recent "mood", a user's profile contains information on her latest activity on the system (songs that she has been listening lately) and the clustering algorithm updates a user's "neighborhood" dynamically. Finally, the last layer in the architecture maintains connectivity with friends through the explicit social network of users. Each user initially connects to her FSTREAMBOX which manages all operations on her behalf. For instance, user d of Figure 1 is connected to her FSTREAMBOX and benefits from social caching from the FSTREAMBOX of both an explicit friend (*i.e.* User b) and an implicit friend on the interest-based topology (*i.e.* User g). Functionalities provided by FSTREAM are explained below.

Social network As depicted in the architecture mentioned above, users can explicitly declare other users as *friends* in FSTREAM. Declaring a user as a *friend*, is equivalent to granting her access to your music collection. The granted access is subject to the sharing preferences on the owner of the collection has defined (described below) and the data remain stored on the owner's FSTREAMBOX while the two users are presented with the merged indexes of their collections. Listening a track generates a stream from the box where it is stored to the user who consumes it (User d on Figure 1).

Ownership and sharing restriction FSTREAM enables users to define their sharing preferences at the granularity of tracks. The full collection or part of it can be *public*, *shared only with its social network* or *private*. *Public* means that anybody can view, listen or copy the content while *private* restricts access to only the owner of the collection. Finally, a user is able to restrict access to her collection *only to her social network*, in this case only users declared as friends can view and listen the content.

Recommendation system As mentioned earlier, one of the objectives of FSTREAM is to discover and connect users that share similar tastes, even in the case that they are not among each other's explicit social network. This is guaranteed by the gossip-based clustering protocol described above that manages to cluster users by exchanging their profiles. Recommendations presented to users are based on the activity of both explicit and implicit *friends*. This functionality permits FSTREAM to work as a recommendation system that "proposes" new content to a given user based on what other users with similar tastes listen. Furthermore, this interest-based topology could be further leveraged in FSTREAM to recommend items to users [1, 2] and to perform the social caching.

Search Users can also launch requests for specific tracks. To answer this need and to locate specific content in the network, FSTREAM benefits from search mechanisms. This mechanism first leverages profile exchanged for the building of overlays. As the profile contains information on the collection, the requested track can be locally present in profile information. Otherwise FSTREAM uses a gossip protocol [3] to localize the content on the system.

Storage Space and Social Caching FSTREAM relies on a self-sufficient storage infrastructure (no need for central infrastructure) where users contribute storage resources to the service. This contributed storage space is divided in two parts, the first one is used to store the user's local collection and is controlled by the user, while the second is used by FSTREAM to guarantee content availability and good Quality-Of-Service (QoS). In this respect, it can be seen as a *distributed social cache*. Its *social* aspect relies on the fact that FSTREAM decides what to store and which item to replace in this part according to the music preferences of the owner of the machine, but also the ones of her *social network*. This differentiates FSTREAM's cache replacement policy from classic ones that apply policies like *Least-Recently-Used* or *Least-Frequently-Used*. The purpose of this social cache is twofold. On one hand, it helps with content availability in the face of node disconnection while on the other, it helps the system to serve popular content and face traffic spikes.

3 Summary

Following the evolution on how users consume music, we present the design of FSTREAM, a distributed social music sharing and streaming service based on FSTREAMBOX hosted at each user. FSTREAM enables users to access to their music collection everywhere and at any time and to share it with friends, search for specific tracks and discover new ones thanks to a recommendation system in a fully decentralized manner. At the same time, FSTREAM provides a fine grained sharing policy that permits the user to have full control over the way she shares her music and manage resource allocation.

References

1. A. Boutet, D. Frey, R. Guerraoui, A. Jégou, and A.-M. Kermarrec. WhatsUp Decentralized Instant News Recommender. In *IPDPS 2013*, Boston, États-Unis, May 2013.
2. J. Carretero, F. Isaila, A.-M. Kermarrec, F. Taïani, and J. Tirado. Geology: Modular Georecommendation in Gossip-Based Social Networks. In *ICDCS*, 2012.
3. D. Kempe, J. Kleinberg, and A. Demers. Spatial gossip and resource location protocols. *J. ACM*, 2004.
4. G. Kreitz and F. Niemela. Spotify – large scale, low latency, p2p music-on-demand streaming. In *P2P*, 2010.
5. M. Marcon, B. Viswanath, M. Cha, and P. Gummadi. Sharing social content from home: a measurement-driven feasibility study. In *NOSSDAV*, 2011.
6. V. Valancius, N. Laoutaris, L. Massoulie, P. Rodriguez, and C. Diot. Greening the Internet with Nano Data Centers. In *CoNEXT*, 2009.
7. S. Voulgaris, D. Gavidia, and M. v. Steen. Cyclon: Inexpensive membership management for unstructured p2p overlays. *J. Network Syst. Manage.*, 2005.
8. S. Voulgaris and M. v. Steen. Epidemic-style management of semantic overlays for content-based searching. In *Euro-Par*, 2005.